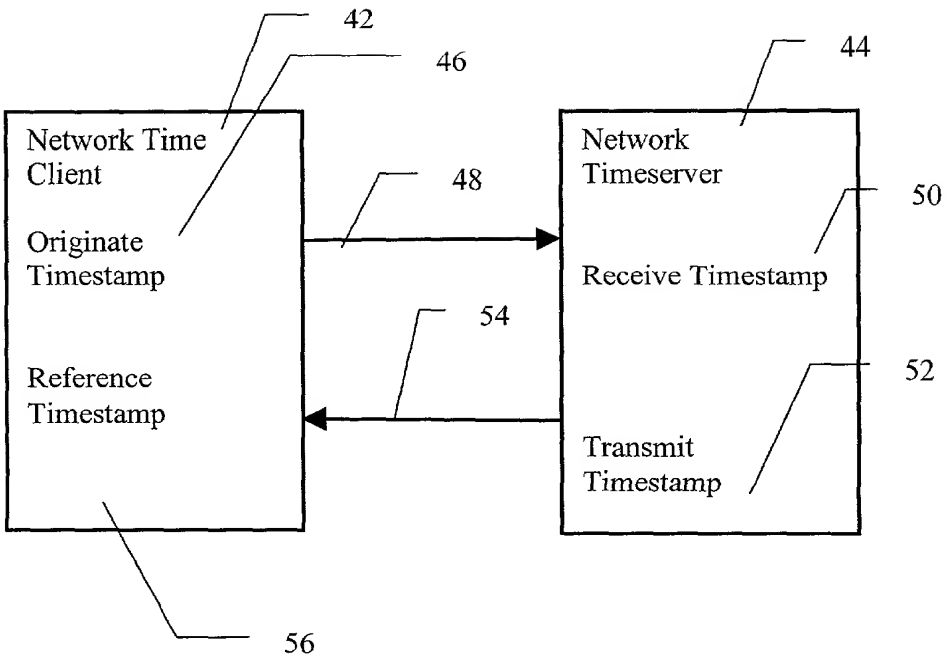
42
46
44
Network Time Client
Originate Timestamp
Reference Timestamp
48
Network Timeserver
50
Receive Timestamp
54
52
Transmit Timestamp
56

Figure 1

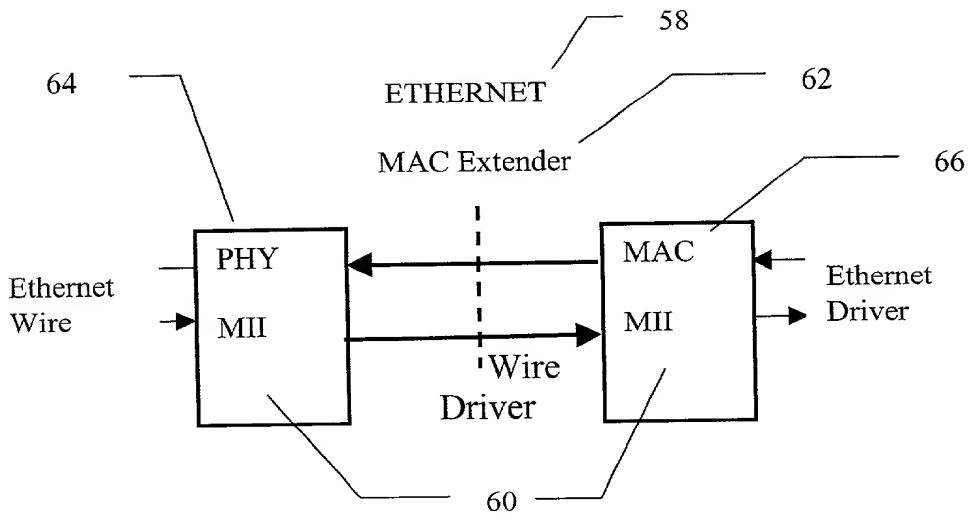
58
64
ETHERNET
62
MAC Extender
66
PHY
MAC
Ethernet
Wire
MII
MII
Ethernet
Driver
Wire
Driver
60

Figure 2

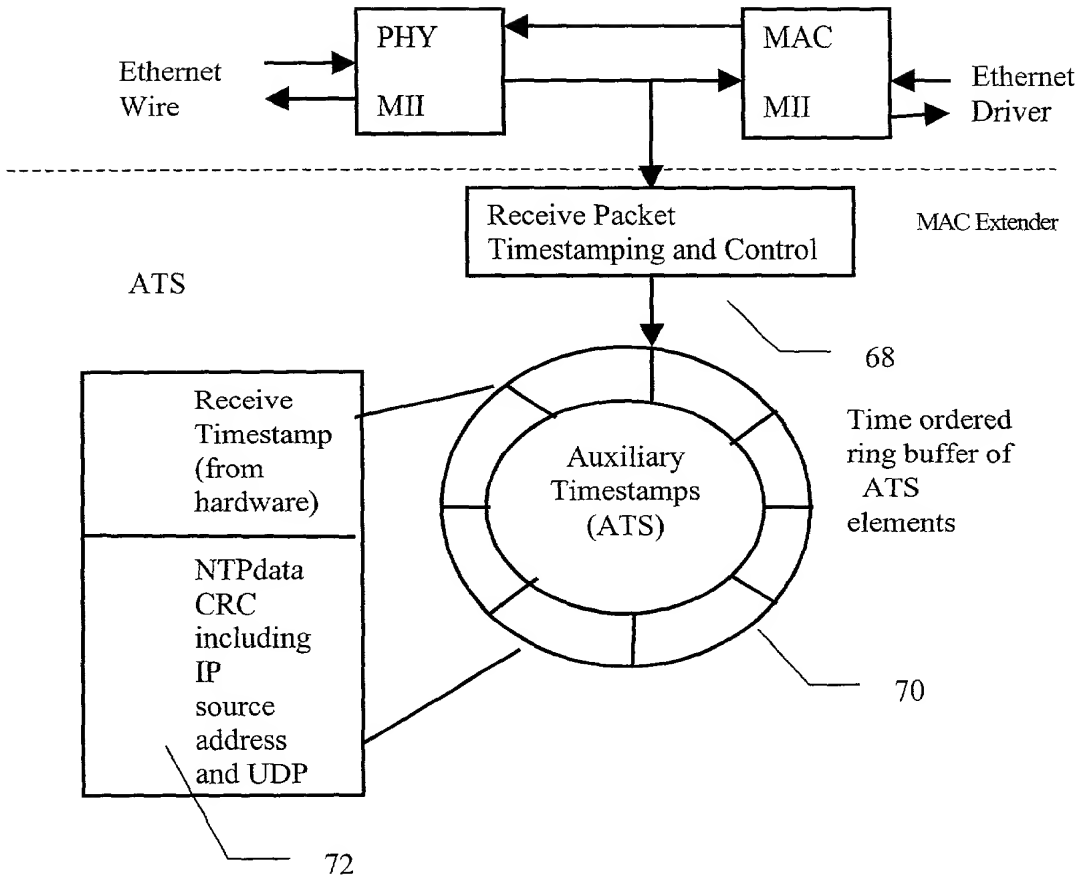
PHY
MII
MAC
MII
Ethernet
Wire
Ethernet
Driver
Receive Packet
Timestamping and Control
MAC Extender
ATS
68
Receive
Timestamp
(from
hardware)
NTPdata
CRC
including
IP
source
address
and UDP
Auxiliary
Timestamps
(ATS)
Time ordered
ring buffer of
ATS
elements
70
72

Figure 3

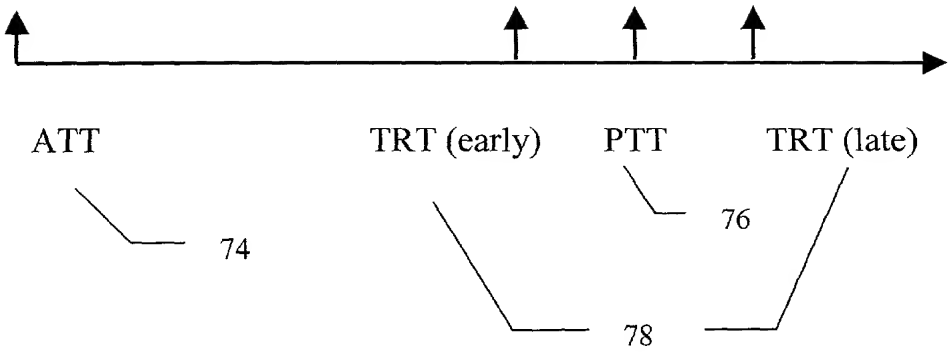

Figure 4

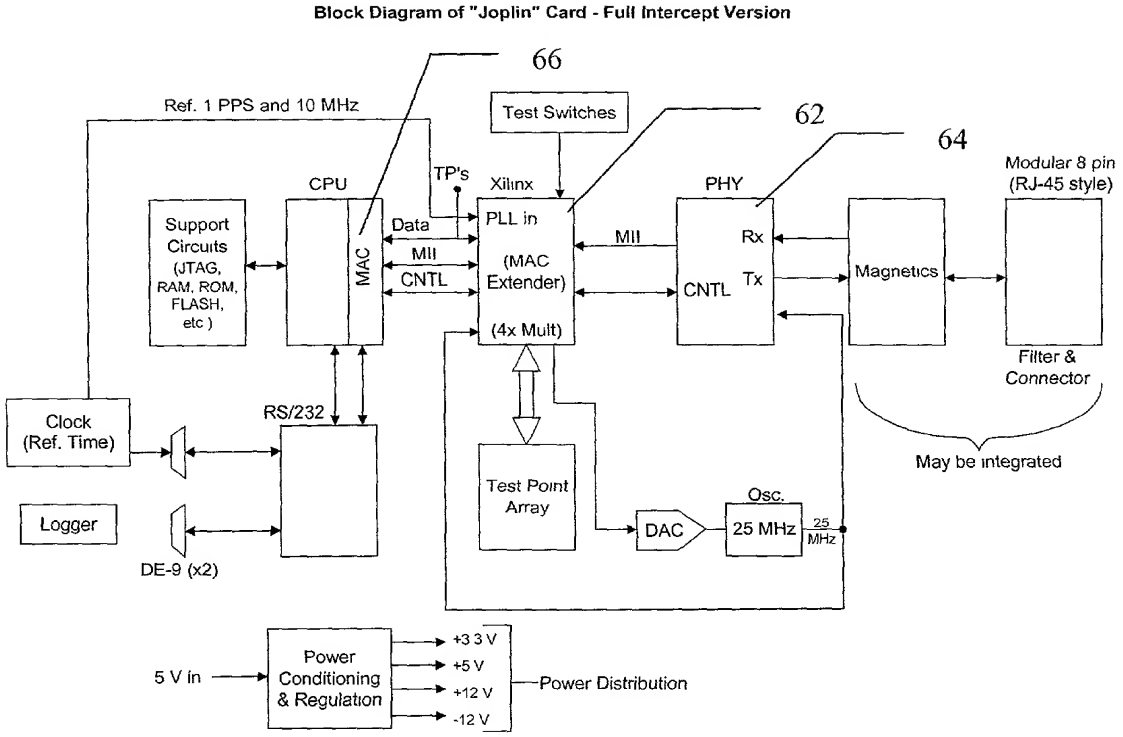
Block Diagram of "Joplin" Card - Full Intercept Version
66
Ref. 1 PPS and 10 MHz
Test Switches
62
64
Modular 8 pin
(RJ-45 style)
CPU
TP's
Xilinx
PHY
Support
Circuits
(JTAG,
RAM, ROM,
FLASH,
etc)
MAC
Data
MII
CNTL
PLL in
(MAC
Extender)
(4x Mult)
MII
Rx
Tx
CNTL
Magnetics
Filter &
Connector
May be integrated
Clock
(Ref. Time)
RS/232
Logger
DE-9 (x2)
Test Point
Array
DAC
Osc.
25 MHz
25
MHz
5 V in
Power
Conditioning
& Regulation
+3 3 V
+5 V
+12 V
-12 V
Power Distribution

Figure 5

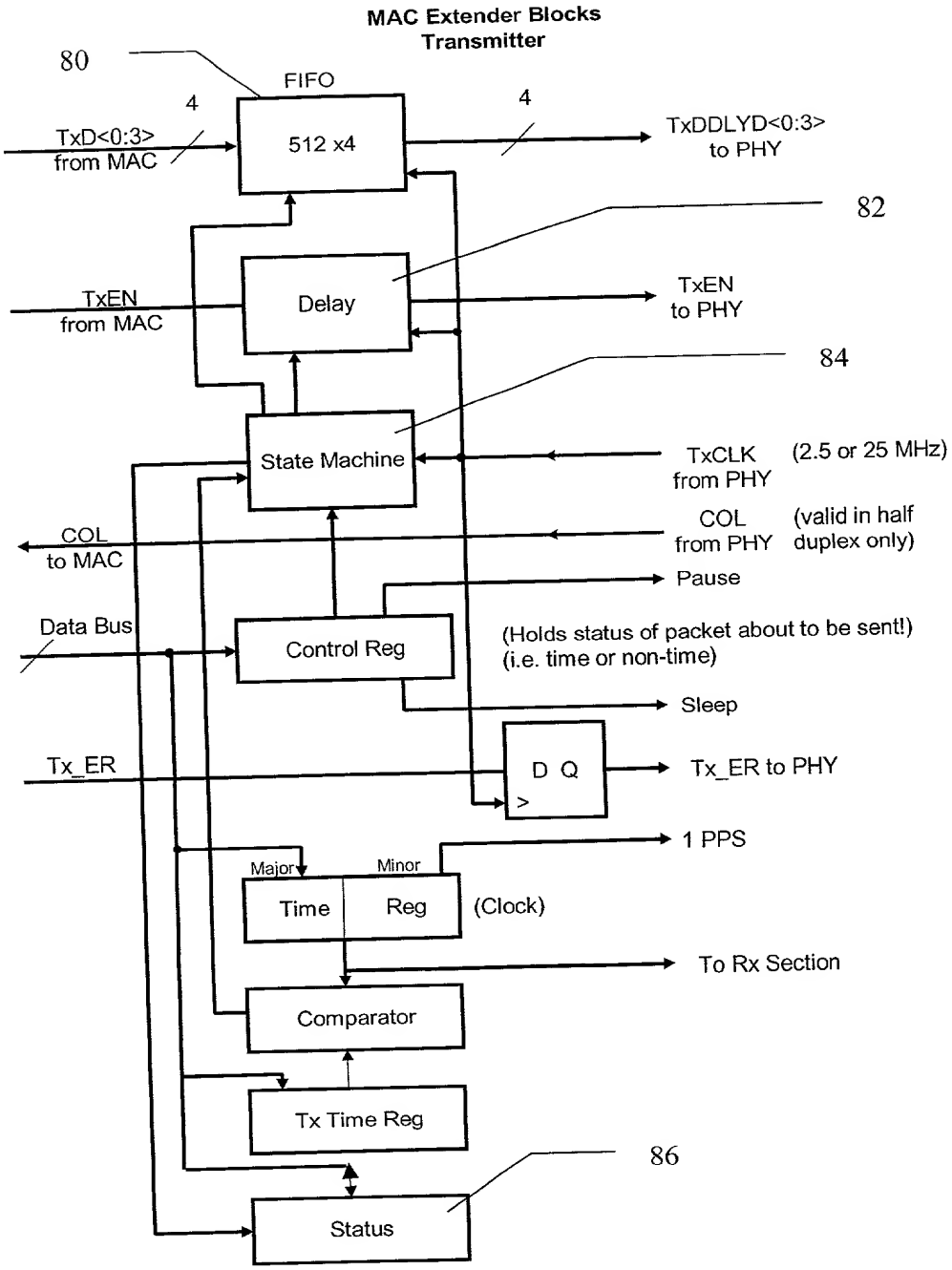
MAC Extender Blocks
Transmitter
80
FIFO
4
4
TxD<0:3>
from MAC
512 x4
TxDDLYD<0:3>
to PHY
82
Delay
TxEN
from MAC
TxEN
to PHY
84
State Machine
TxCLK
from PHY
(2.5 or 25 MHz)
COL
to MAC
COL
from PHY
(valid in half
duplex only)
Pause
Data Bus
Control Reg
(Holds status of packet about to be sent!)
(i.e. time or non-time)
Sleep
Tx_ER
D Q
>
Tx_ER to PHY
1 PPS
Major
Minor
Time
Reg
(Clock)
To Rx Section
Comparator
Tx Time Reg
86
Status

Figure 6

MAC Extender Receiver

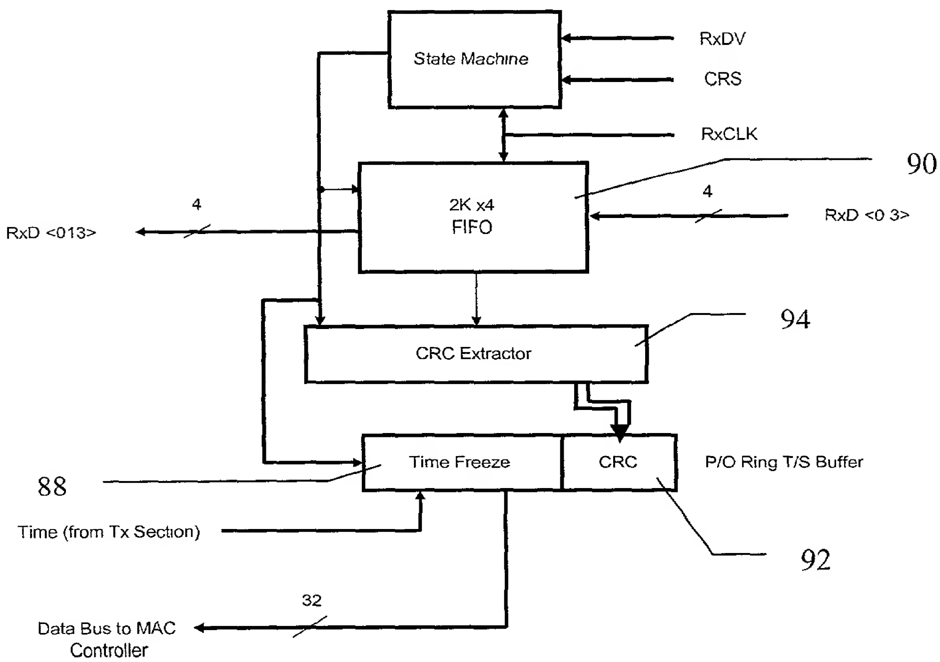

Figure 7

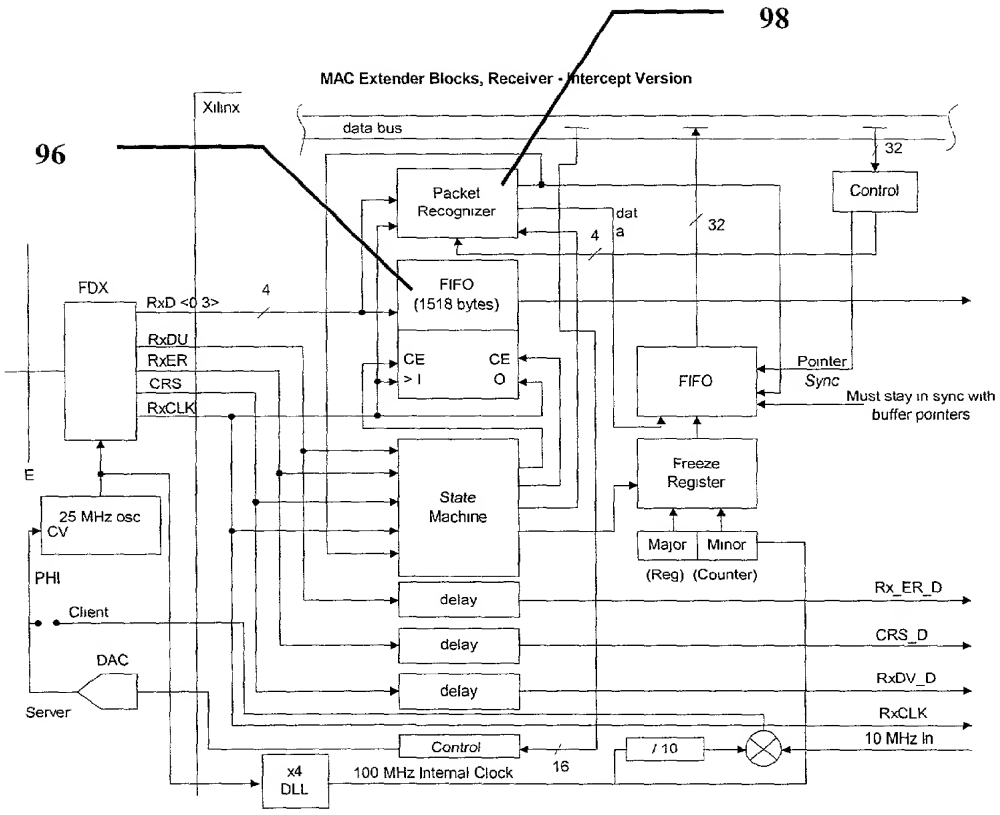
98
MAC Extender Blocks, Receiver - Intercept Version
96
Xilinx
data bus
32
Control
Packet Recognizer
dat a
4
32
FIFO (1518 bytes)
FDX
RxD <0 3>
4
RxDU
RxER
CRS
RxCLK
CE > I
CE O
FIFO
Pointer Sync
Must stay in sync with buffer pointers
E
State Machine
Freeze Register
25 MHz osc
CV
PHI
Major
Minor
(Reg)
(Counter)
Client
delay
Rx_ER_D
DAC
delay
CRS_D
Server
delay
RxDV_D
RxCLK
Control
16
/ 10
10 MHz In
x4 DLL
100 MHz Internal Clock

Figure 8

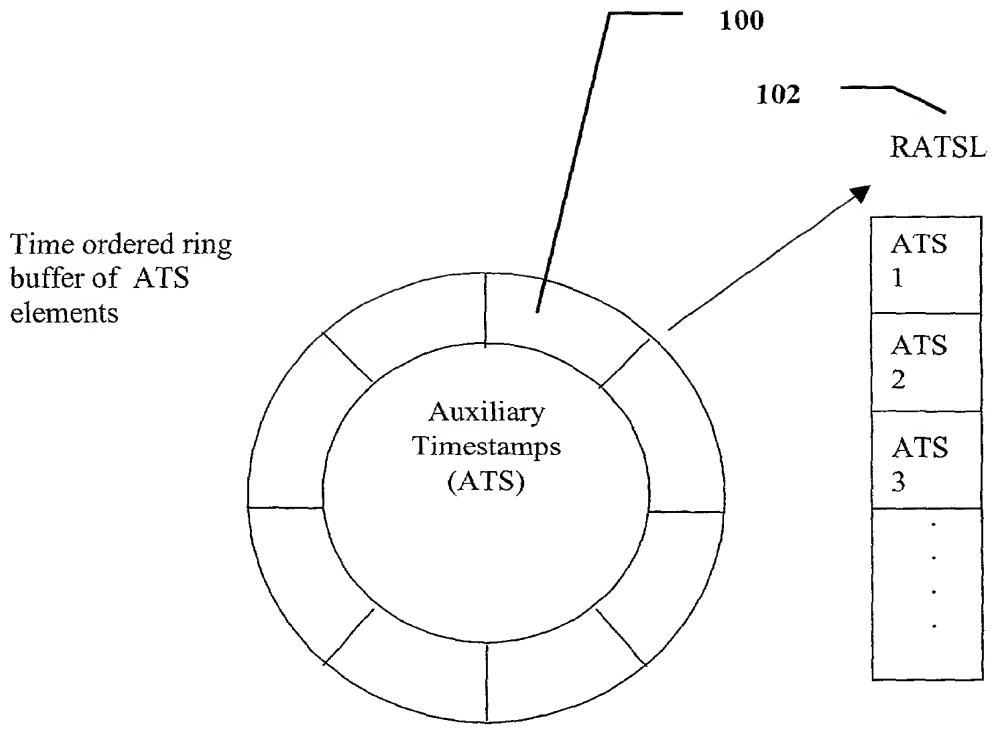
100
102
RATSL
Time ordered ring buffer of ATS elements
Auxiliary Timestamps (ATS)
ATS 1
ATS 2
ATS 3

Figure 9

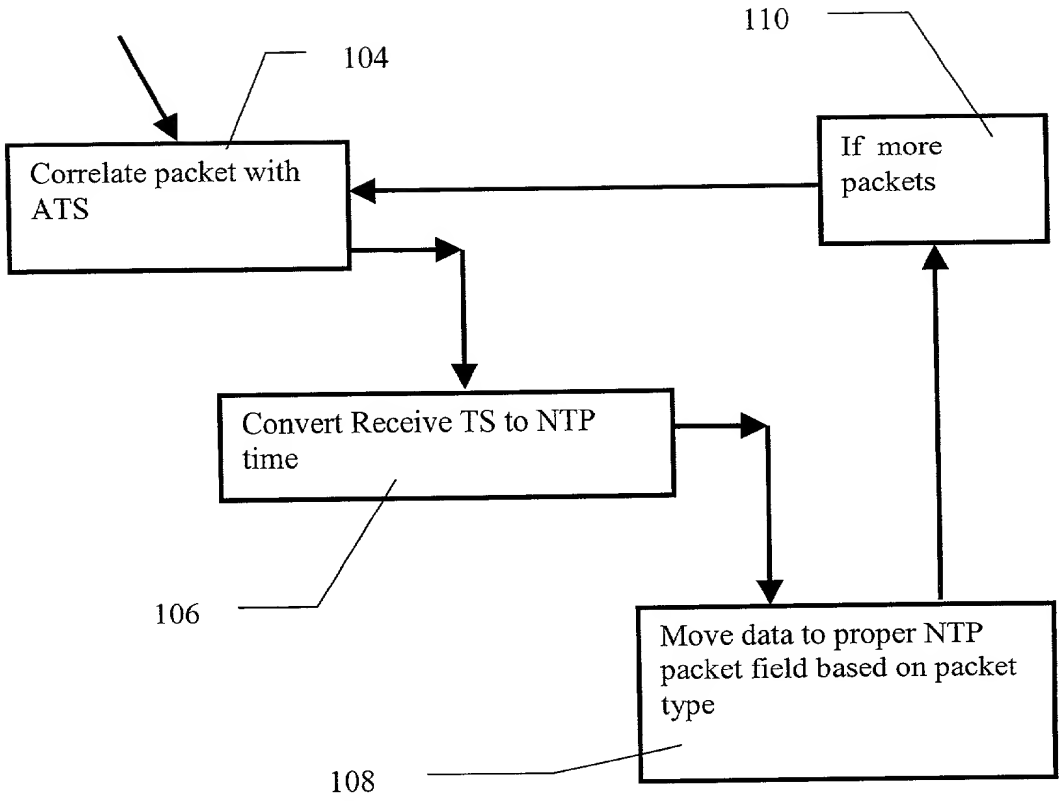
110
104
Correlate packet with ATS
If more packets
Convert Receive TS to NTP time
106
Move data to proper NTP packet field based on packet type
108

Figure 10

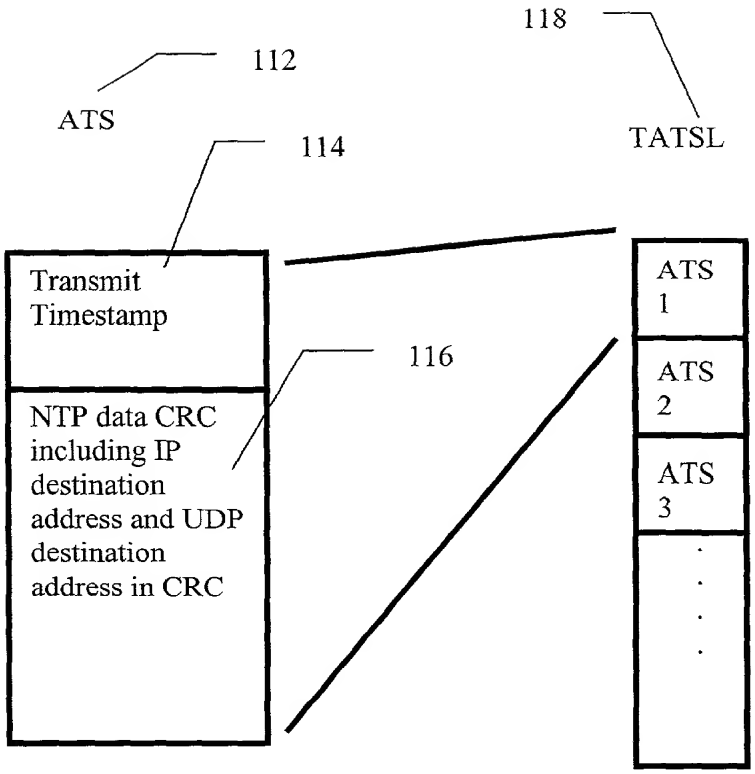
118
112
ATS
114
TATSL
Transmit Timestamp
116
NTP data CRC including IP destination address and UDP destination address in CRC
ATS 1
ATS 2
ATS 3

Figure 11

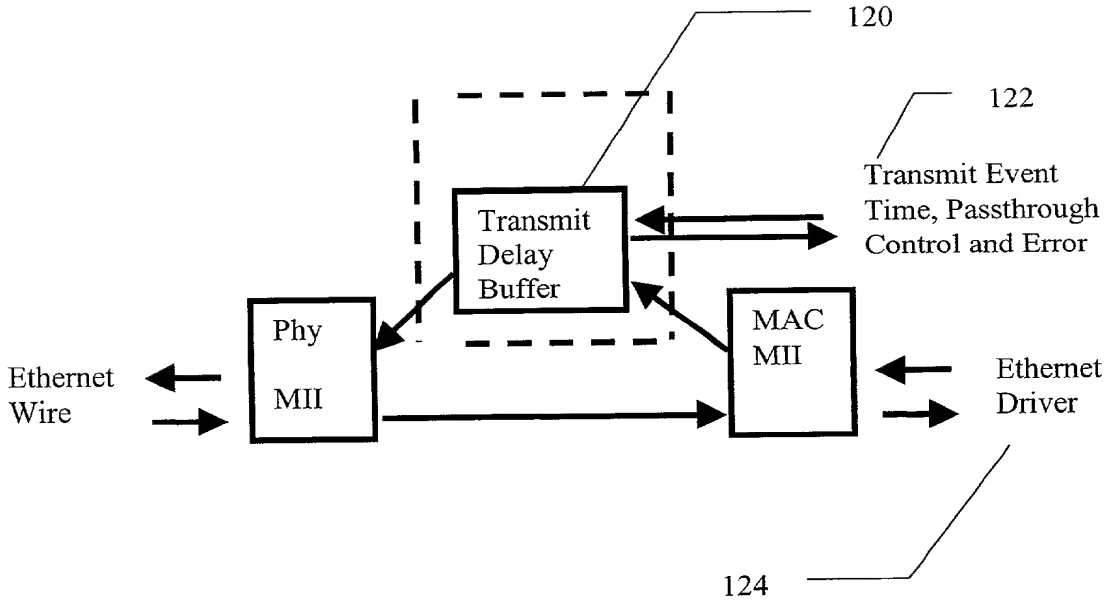

Figure 12

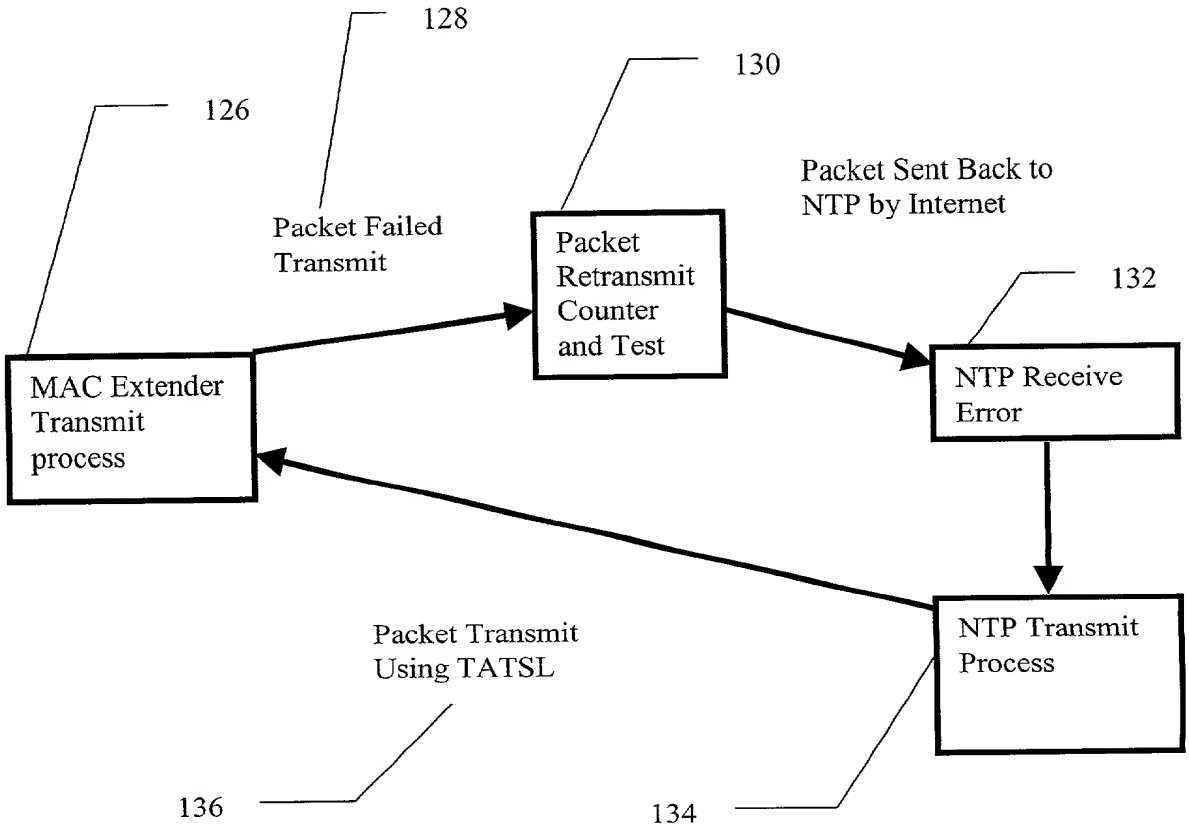
128
130
126
Packet Sent Back to
NTP by Internet
Packet Failed
Transmit
Packet
Retransmit
Counter
and Test
132
MAC Extender
Transmit
process
NTP Receive
Error
NTP Transmit
Process
Packet Transmit
Using TATSL
136
134

Figure 13

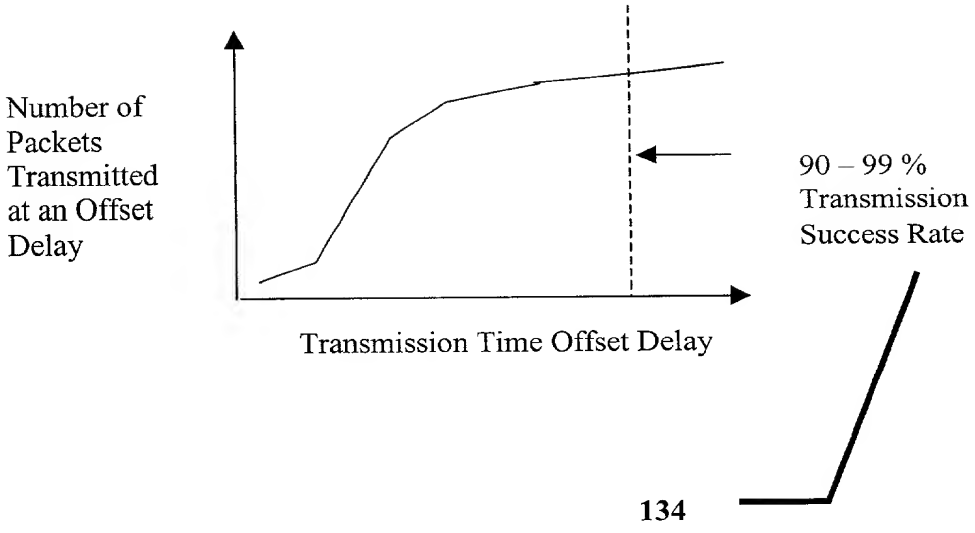
Number of Packets Transmitted at an Offset Delay
Transmission Time Offset Delay
90 – 99 % Transmission Success Rate
134

Figure 14

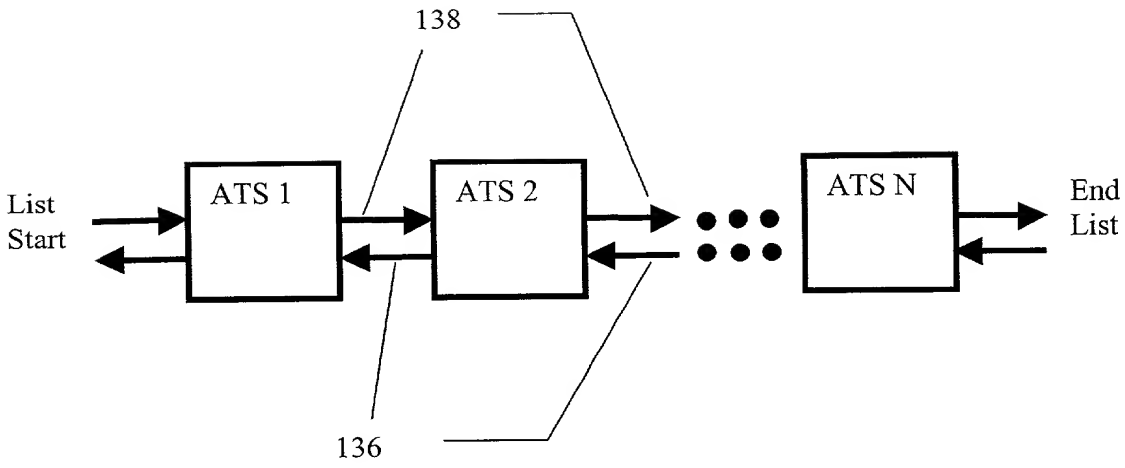

Figure 15

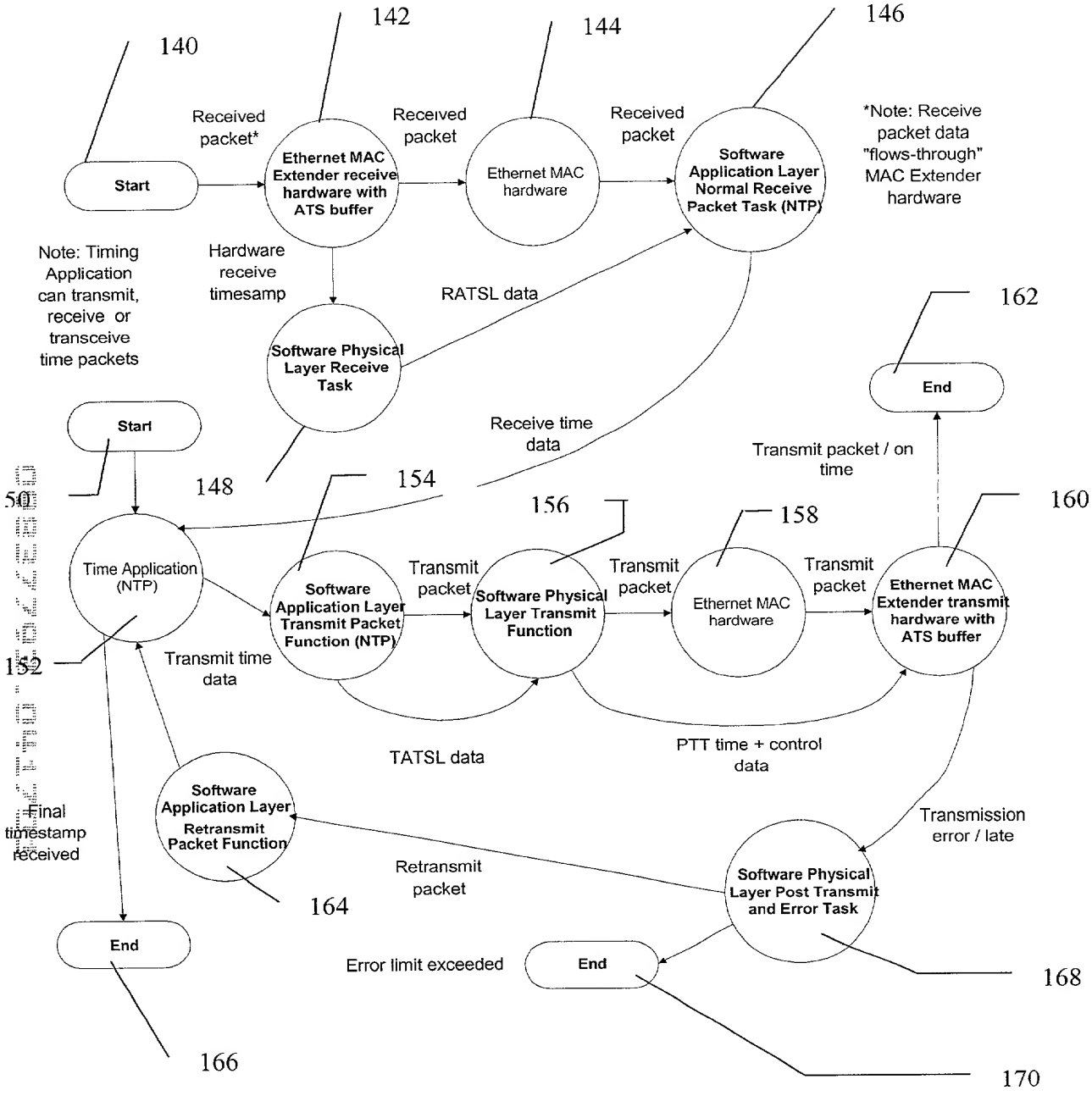
140
142
144
146
Start
Received packet*
Ethernet MAC Extender receive hardware with ATS buffer
Received packet
Ethernet MAC hardware
Received packet
Software Application Layer Normal Receive Packet Task (NTP)
*Note: Receive packet data "flows-through" MAC Extender hardware
Note: Timing Application can transmit, receive or transceive time packets
Hardware receive timesamp
RATSL data
Software Physical Layer Receive Task
162
End
Start
Receive time data
Transmit packet / on time
148
154
156
158
160
150
Time Application (NTP)
Software Application Layer Transmit Packet Function (NTP)
Transmit packet
Software Physical Layer Transmit Function
Transmit packet
Ethernet MAC hardware
Transmit packet
Ethernet MAC Extender transmit hardware with ATS buffer
152
Transmit time data
TATSL data
PTT time + control data
Final timestamp received
Software Application Layer Retransmit Packet Function
Transmission error / late
Retransmit packet
Software Physical Layer Post Transmit and Error Task
164
End
Error limit exceeded
End
168
166
170

Figure 16

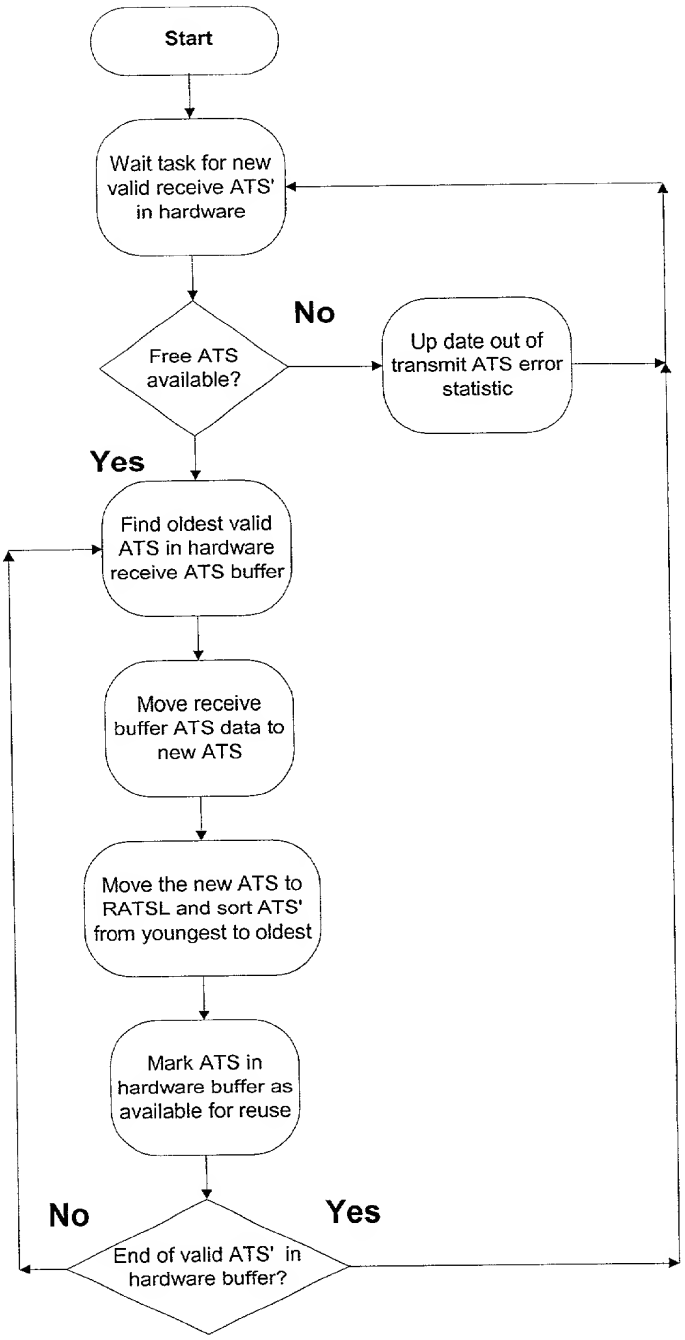
Start
Wait task for new valid receive ATS' in hardware
Free ATS available?
No
Up date out of transmit ATS error statistic
Yes
Find oldest valid ATS in hardware receive ATS buffer
Move receive buffer ATS data to new ATS
Move the new ATS to RATSL and sort ATS' from youngest to oldest
Mark ATS in hardware buffer as available for reuse
No
Yes
End of valid ATS' in hardware buffer?

Figure 17

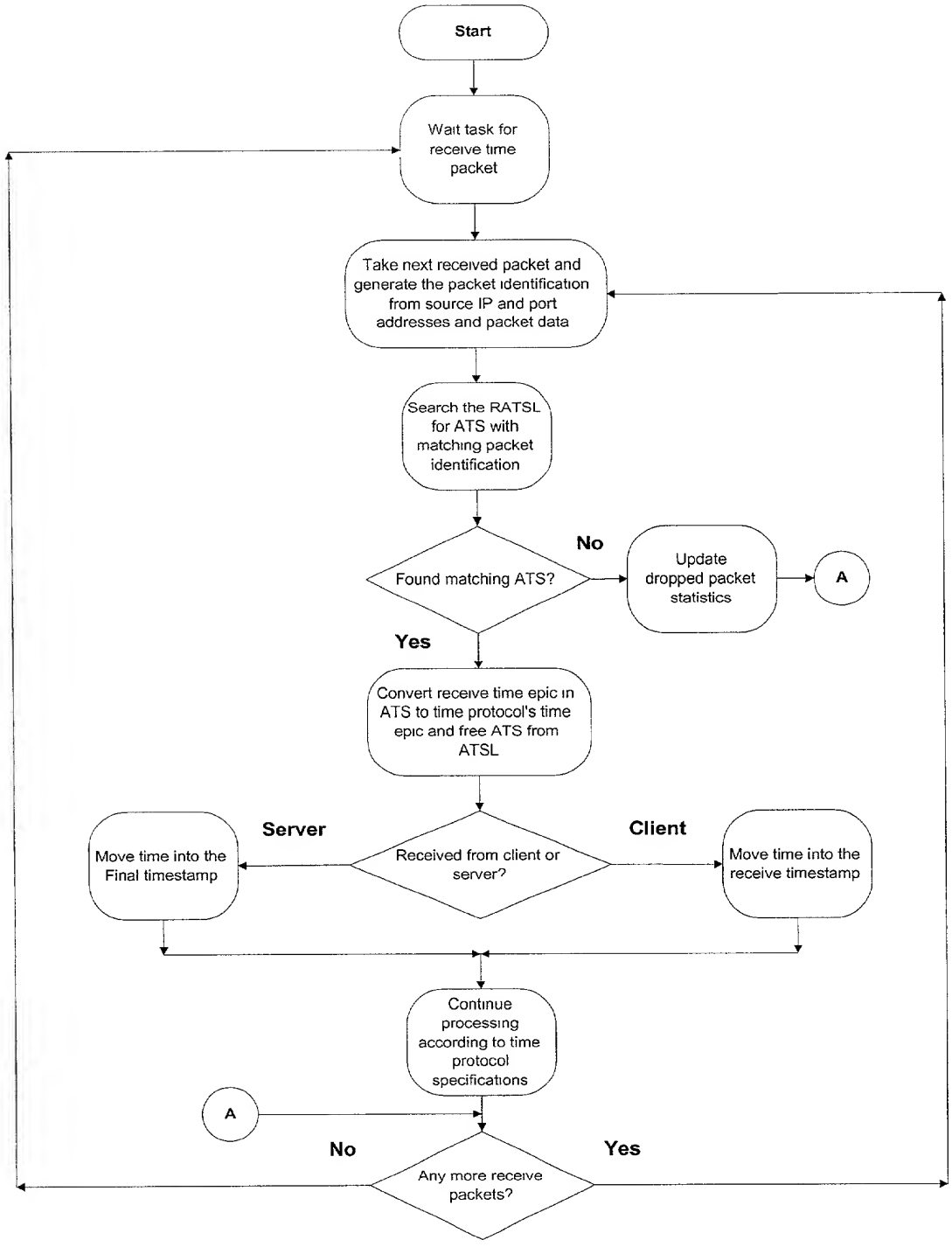
Start
Wait task for receive time packet
Take next received packet and generate the packet identification from source IP and port addresses and packet data
Search the RATSL for ATS with matching packet identification
Found matching ATS?
No
Update dropped packet statistics
A
Yes
Convert receive time epic in ATS to time protocol's time epic and free ATS from ATSL
Server
Received from client or server?
Client
Move time into the Final timestamp
Move time into the receive timestamp
Continue processing according to time protocol specifications
A
No
Any more receive packets?
Yes

Figure 18

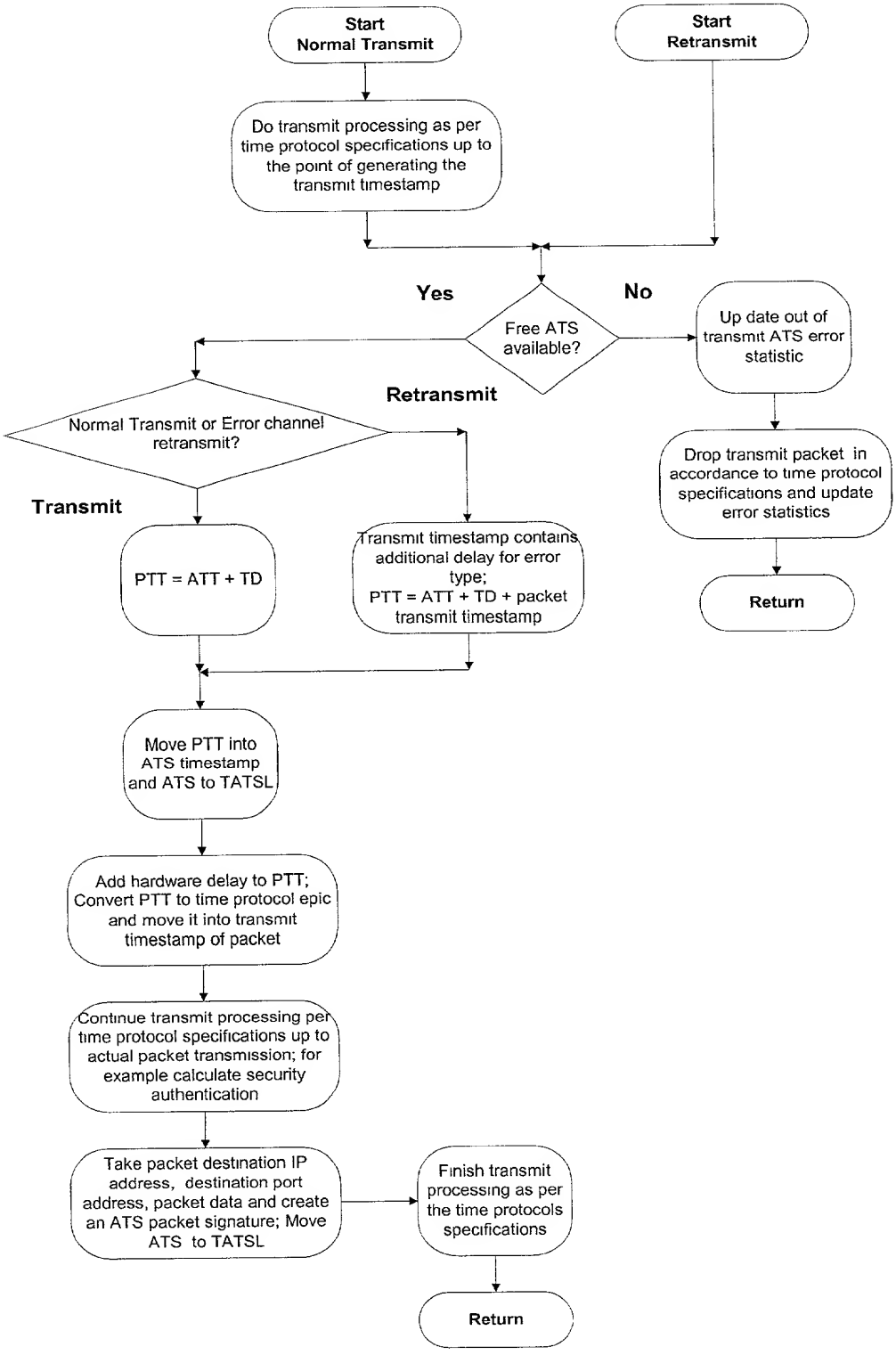
Start
Normal Transmit
Start
Retransmit
Do transmit processing as per time protocol specifications up to the point of generating the transmit timestamp
Yes
No
Free ATS available?
Up date out of transmit ATS error statistic
Retransmit
Normal Transmit or Error channel retransmit?
Drop transmit packet in accordance to time protocol specifications and update error statistics
Transmit
PTT = ATT + TD
Transmit timestamp contains additional delay for error type;
PTT = ATT + TD + packet transmit timestamp
Return
Move PTT into ATS timestamp and ATS to TATSL
Add hardware delay to PTT; Convert PTT to time protocol epic and move it into transmit timestamp of packet
Continue transmit processing per time protocol specifications up to actual packet transmission; for example calculate security authentication
Take packet destination IP address, destination port address, packet data and create an ATS packet signature; Move ATS to TATSL
Finish transmit processing as per the time protocols specifications
Return

Figure 19

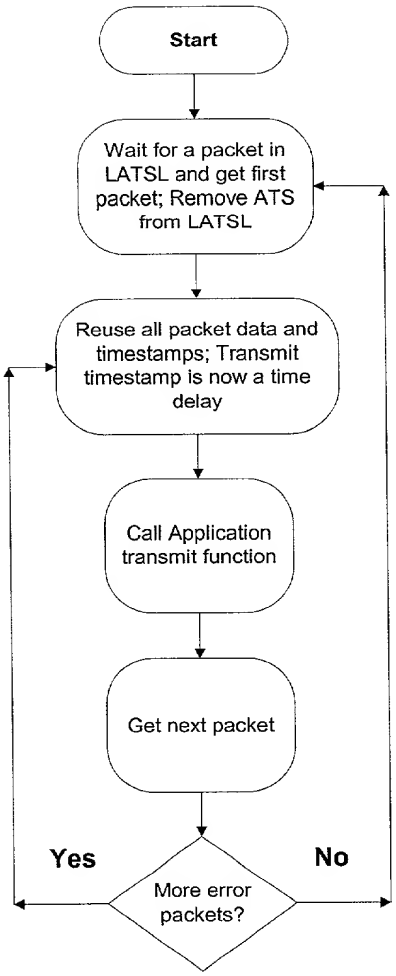
Start
Wait for a packet in LATSL and get first packet; Remove ATS from LATSL
Reuse all packet data and timestamps; Transmit timestamp is now a time delay
Call Application transmit function
Get next packet
Yes
No
More error packets?

Figure 20

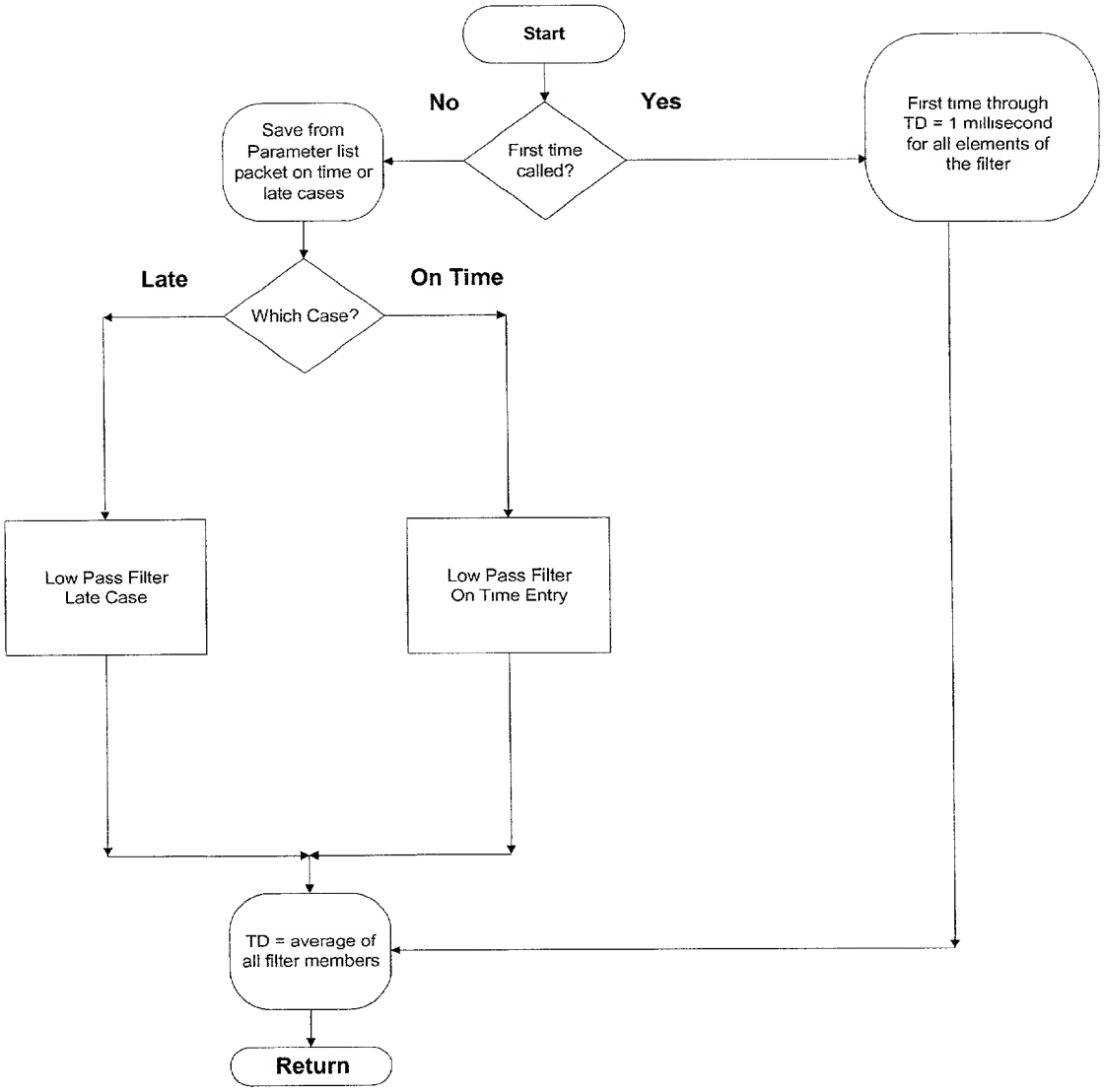
Start
No
Yes
Save from Parameter list packet on time or late cases
First time called?
First time through TD = 1 millisecond for all elements of the filter
Late
On Time
Which Case?
Low Pass Filter Late Case
Low Pass Filter On Time Entry
TD = average of all filter members
Return

Figure 21

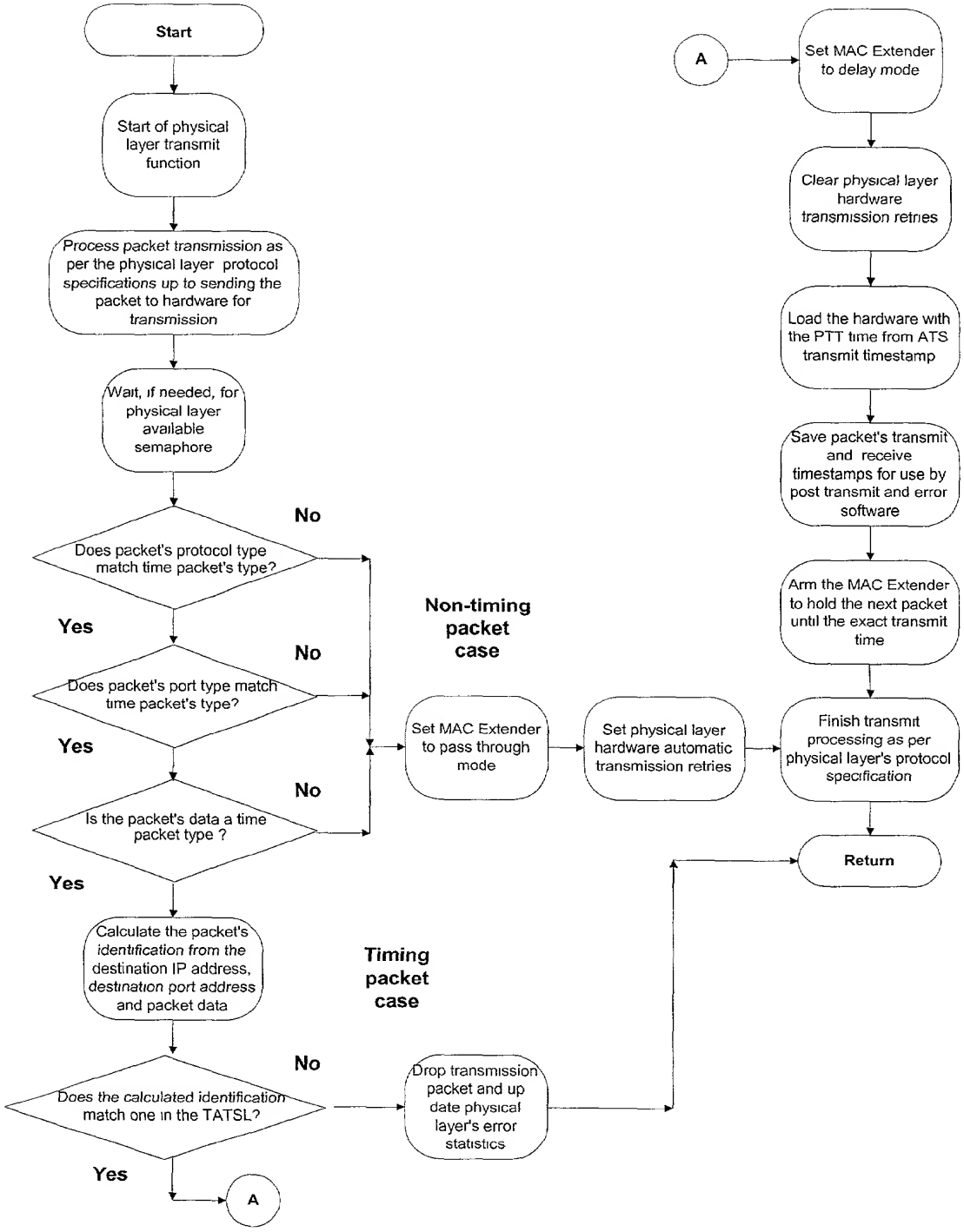
Start
Start of physical layer transmit function
Process packet transmission as per the physical layer protocol specifications up to sending the packet to hardware for transmission
Wait, if needed, for physical layer available semaphore
Does packet's protocol type match time packet's type?
No
Yes
Does packet's port type match time packet's type?
No
Yes
Is the packet's data a time packet type ?
No
Yes
Calculate the packet's identification from the destination IP address, destination port address and packet data
Does the calculated identification match one in the TATSL?
No
Yes
A
Non-timing packet case
Timing packet case
Set MAC Extender to pass through mode
Drop transmission packet and up date physical layer's error statistics
Set physical layer hardware automatic transmission retries
A
Set MAC Extender to delay mode
Clear physical layer hardware transmission retries
Load the hardware with the PTT time from ATS transmit timestamp
Save packet's transmit and receive timestamps for use by post transmit and error software
Arm the MAC Extender to hold the next packet until the exact transmit time
Finish transmit processing as per physical layer's protocol specification
Return

Figure 22

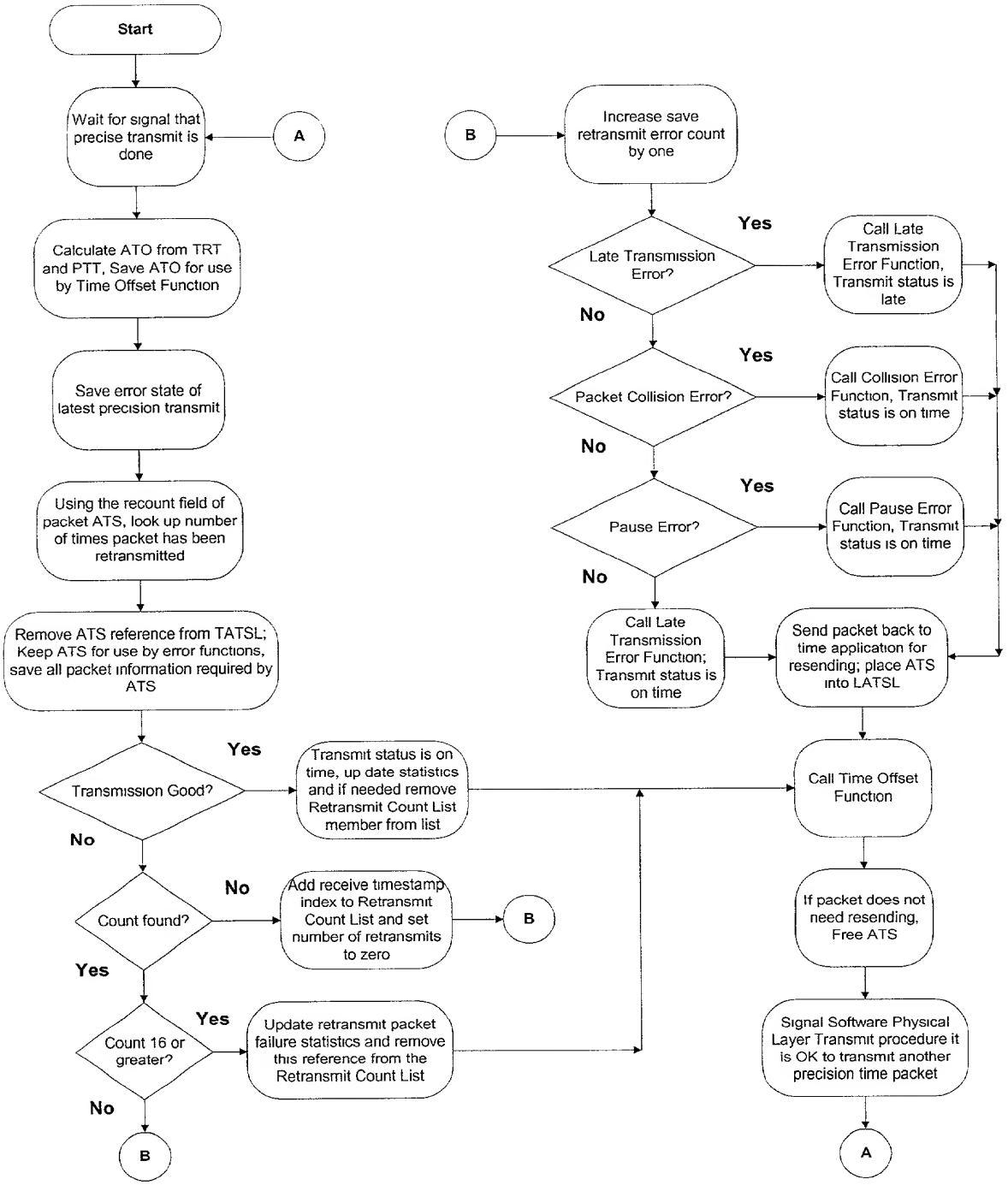
Start
Wait for signal that precise transmit is done
A
Calculate ATO from TRT and PTT, Save ATO for use by Time Offset Function
Save error state of latest precision transmit
Using the recount field of packet ATS, look up number of times packet has been retransmitted
Remove ATS reference from TATSL; Keep ATS for use by error functions, save all packet information required by ATS
Transmission Good?
Yes
Transmit status is on time, up date statistics and if needed remove Retransmit Count List member from list
No
Count found?
No
Add receive timestamp index to Retransmit Count List and set number of retransmits to zero
B
Yes
Count 16 or greater?
Yes
Update retransmit packet failure statistics and remove this reference from the Retransmit Count List
No
B
B
Increase save retransmit error count by one
Late Transmission Error?
Yes
Call Late Transmission Error Function, Transmit status is late
No
Packet Collision Error?
Yes
Call Collision Error Function, Transmit status is on time
No
Pause Error?
Yes
Call Pause Error Function, Transmit status is on time
No
Call Late Transmission Error Function; Transmit status is on time
Send packet back to time application for resending; place ATS into LATSL
Call Time Offset Function
If packet does not need resending, Free ATS
Signal Software Physical Layer Transmit procedure it is OK to transmit another precision time packet
A

Figure 23

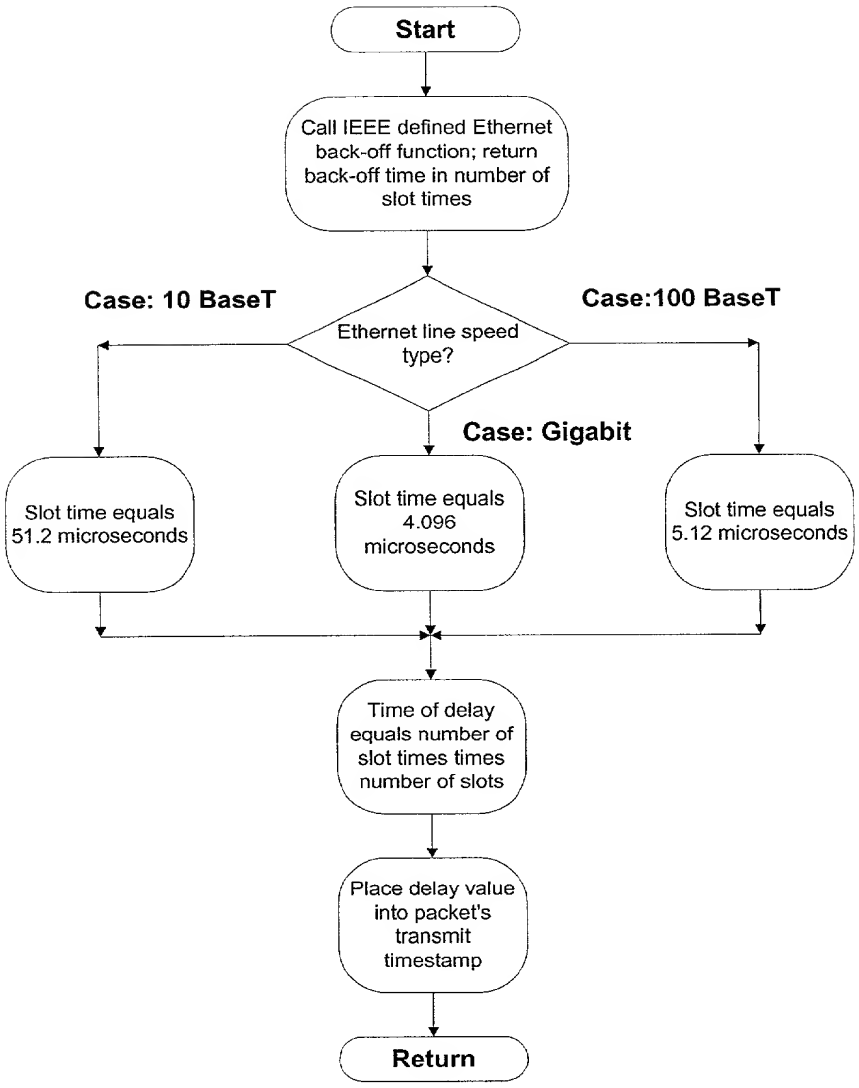
Start
Call IEEE defined Ethernet back-off function; return back-off time in number of slot times
Case: 10 BaseT
Ethernet line speed type?
Case:100 BaseT
Case: Gigabit
Slot time equals 51.2 microseconds
Slot time equals 4.096 microseconds
Slot time equals 5.12 microseconds
Time of delay equals number of slot times times number of slots
Place delay value into packet's transmit timestamp
Return

Figure 24

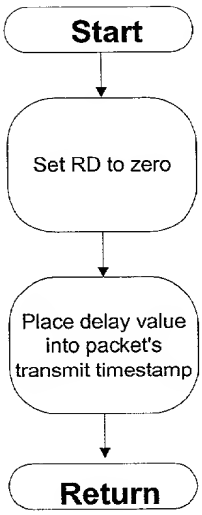
Start
Set RD to zero
Place delay value into packet's transmit timestamp
Return

Figure 25

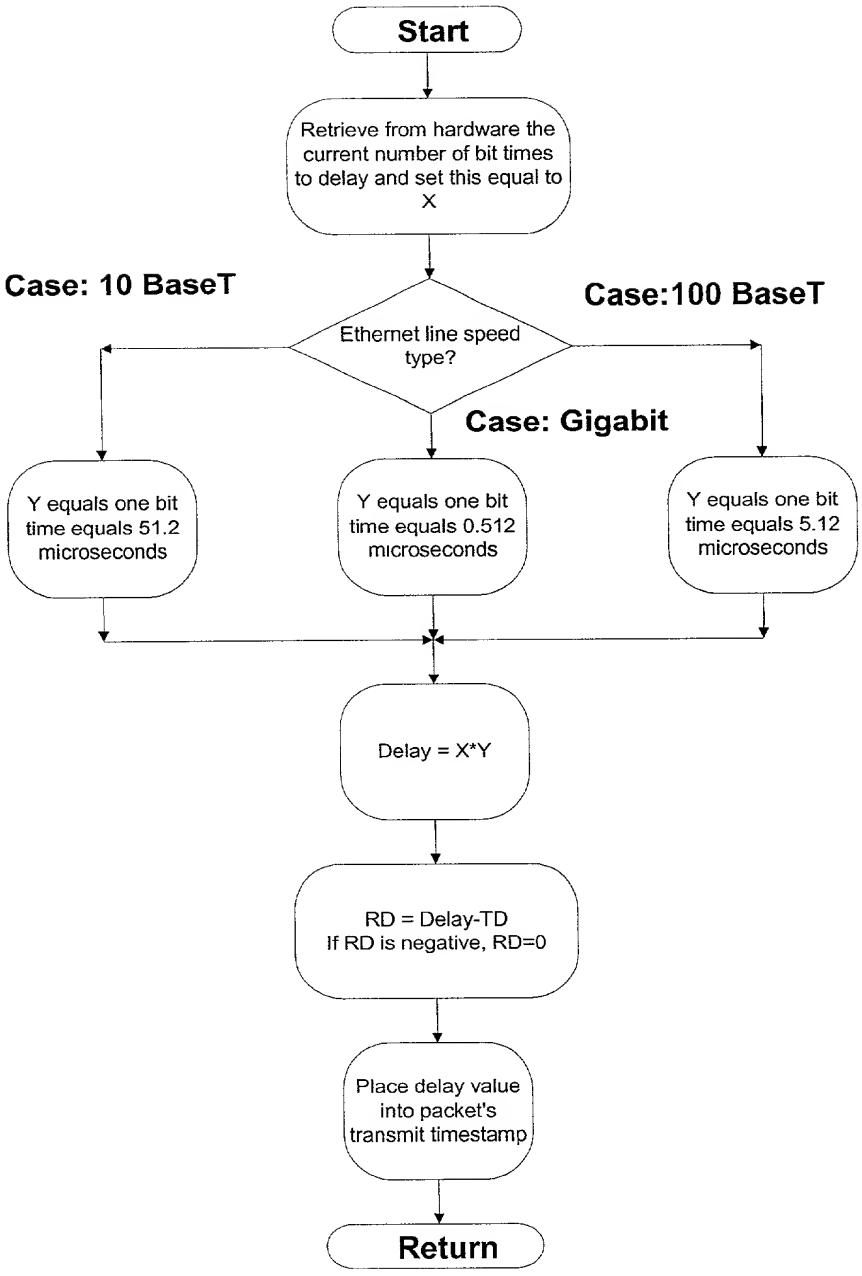
Start
Retrieve from hardware the current number of bit times to delay and set this equal to X
Case: 10 BaseT
Case:100 BaseT
Ethernet line speed type?
Case: Gigabit
Y equals one bit time equals 51.2 microseconds
Y equals one bit time equals 0.512 microseconds
Y equals one bit time equals 5.12 microseconds
Delay = X*Y
RD = Delay-TD
If RD is negative, RD=0
Place delay value into packet's transmit timestamp
Return

Figure 26

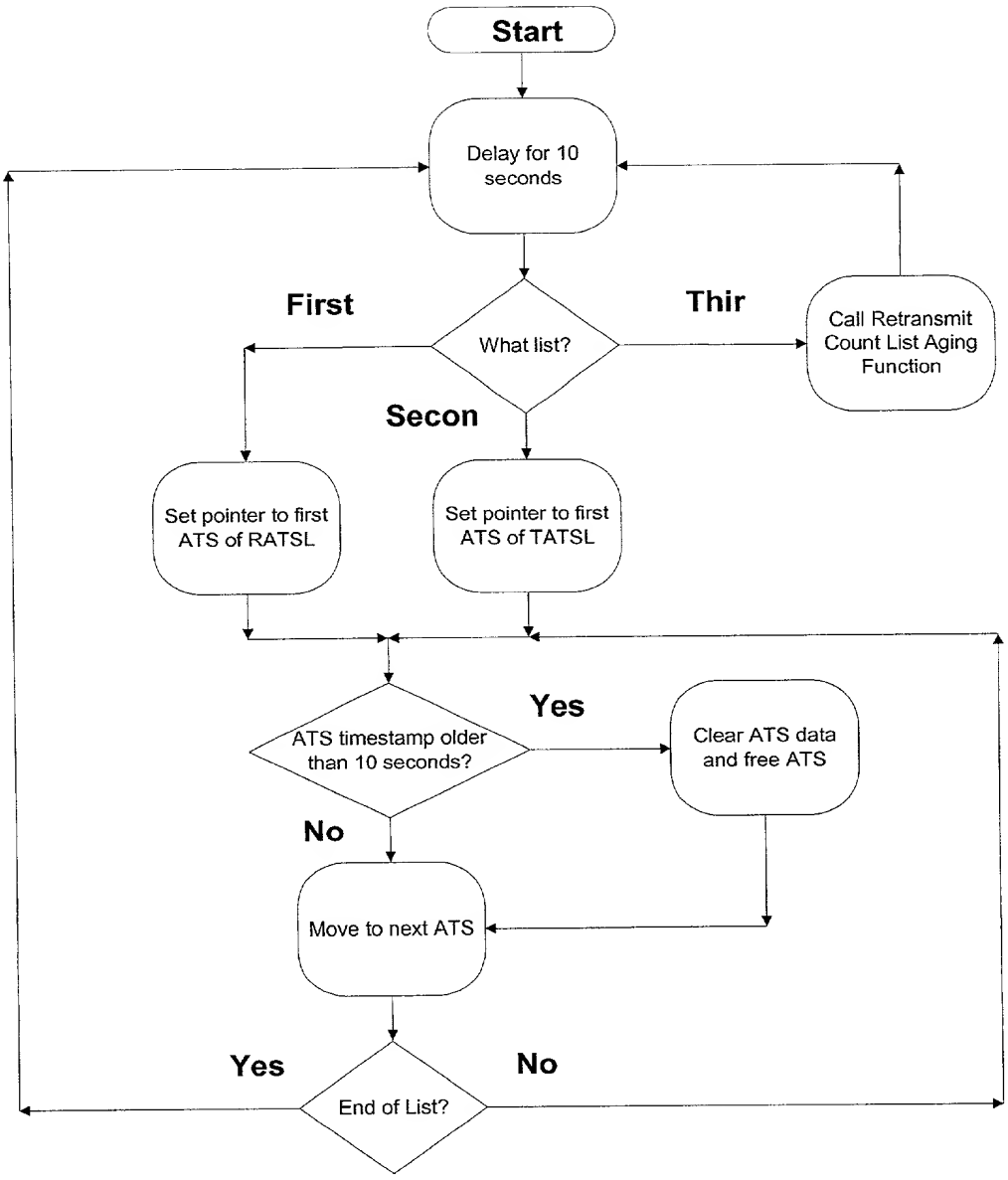
Start
Delay for 10 seconds
First
What list?
Thir
Call Retransmit Count List Aging Function
Secon
Set pointer to first ATS of RATSL
Set pointer to first ATS of TATSL
ATS timestamp older than 10 seconds?
Yes
Clear ATS data and free ATS
No
Move to next ATS
Yes
End of List?
No

Figure 27

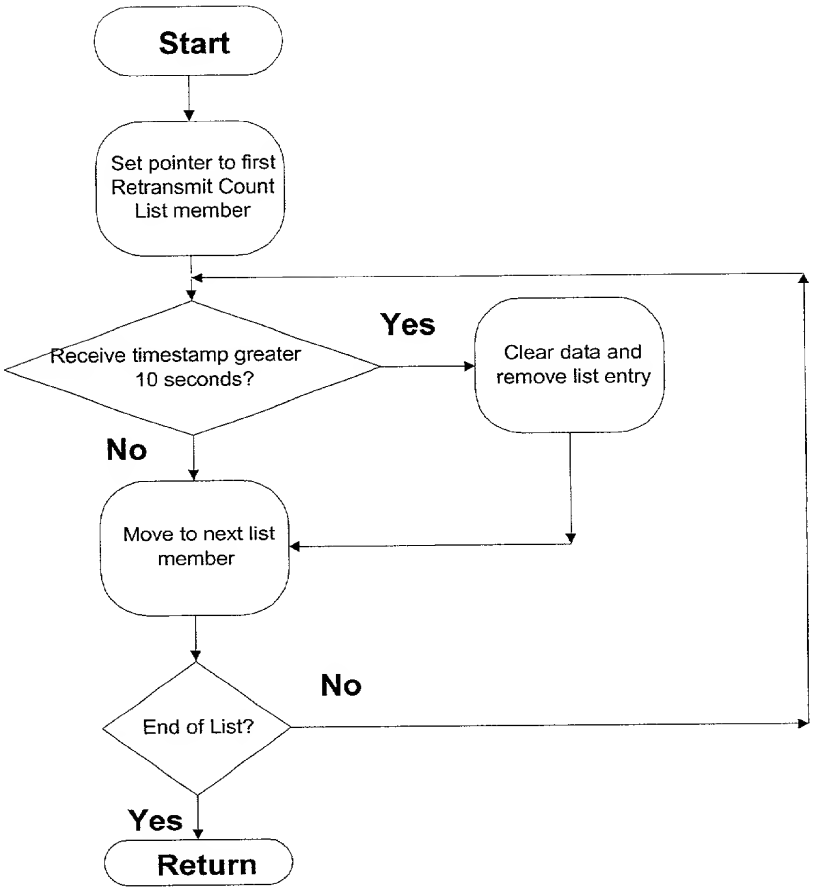
Start
Set pointer to first Retransmit Count List member
Receive timestamp greater 10 seconds?
Yes
Clear data and remove list entry
No
Move to next list member
End of List?
No
Yes
Return

Figure 28